

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the Application.

1 – 7. (cancelled)

8. (currently amended) ~~The computer system of claim 7,~~ A computer system that resolves name collisions by providing type support for multiple type definitions, comprising:

an interface repository including:

a repository naming context; and

a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least one interface definition language declaration, the prefix naming context being adapted to resolve names subordinate to the repository naming context; and

an interface repository loader that accepts as input parameters a specified interface definition language file containing at least one interface definition language declaration, and a specified prefix name, and installs the at least one interface definition language declaration in a prefix naming context having the prefix naming context in the interface repository, wherein the interface repository loader creates a data file identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context.

9. (currently amended) ~~The computer system of claim 7,~~ A computer system that resolves name collisions by providing type support for multiple type definitions, comprising:

an interface repository including:

a repository naming context; and

a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least one interface definition language declaration, the prefix naming context being adapted to resolve names subordinate to the repository naming context; and  
an interface repository loader that accepts as input parameters a specified interface definition language file containing at least one interface definition language declaration, and a specified prefix name, and installs the at least one interface definition language declaration in a prefix naming context having the prefix naming context in the interface repository, wherein the interface repository loader creates the specified prefix naming context in the interface repository if the specified prefix naming context does not exist therein.

B<sup>2</sup> 10. (cancelled)

11. (currently amended) ~~The computer system of claim 10,~~ A computer system that resolves name collisions by providing type support for multiple type definitions, comprising:

an interface repository including:

a repository naming context; and

a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least one interface definition language declaration, the prefix naming context being adapted to resolve names subordinate to the repository naming context;

a memory device that stores the interface repository; and

a processing unit that executes operations of the interface repository loader, wherein the processing unit further executes the interface repository loader to create a data file identified as related to the specified interface definition

language file, and containing an identification of the specified prefix naming context.

12 – 20. (cancelled)

21. (currently amended) ~~The method of claim 19, further comprising the step of:~~ A method of providing type support for multiple type definitions, comprising the steps of:

defining in an interface repository a prefix naming context;

storing the prefix naming context subordinate to a repository naming context in the interface repository, the prefix naming context forming an interface definition language root context for interface definition objects subordinate to the prefix naming context;

specifying an interface definition language file containing at least one interface definition language declaration; specifying a prefix naming context;

storing each interface definition language declaration in the specified interface definition language file into the specified prefix naming context; and

creating a data file identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context.

22 – 23. (cancelled)

24. (currently amended) ~~The method of claim 21, further comprising the step of:~~ A method of providing type support for multiple type definitions, comprising the steps of:

defining in an interface repository a prefix naming context;

storing the prefix naming context subordinate to a repository naming context in the interface repository, the prefix naming context forming an interface definition language root context for interface definition objects subordinate to the prefix naming context;

specifying an interface definition language file containing at least one interface definition language declaration; specifying a prefix naming context;

storing each interface definition language declaration in the specified interface definition language file into the specified prefix naming context;  
creating a data file identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context;  
providing a memory device that stores the at least one client object, and the at least one server object; and  
providing a processing unit that executes a server object in response to an invocation of the server object by a client object.

25. (currently amended) A computer system that resolves name collisions by providing type support for multiple type definitions, comprising:

an interface repository including:

a repository naming context; and

a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least one interface definition language declaration, the prefix naming context being adapted to resolve names subordinate to the repository naming context; and

an interface repository loader that accepts as input parameters a specified interface definition language file containing at least one interface definition language declaration, and a specified prefix name, and installs the at least one interface definition language declaration in a prefix naming context having the prefix naming context in the interface repository.

26. (previously presented) The system of claim 25 wherein the prefix naming context further includes:

at least one naming context defined by an interface definition object and subordinate to the prefix naming context.

27. (previously presented) The computer system of claim 26 wherein at least one interface definition object has a fully scoped object name including a prefix name of the prefix naming context to which the interface definition object is subordinated.

28. (previously presented) The computer system of claim 25 wherein the prefix naming context is immediately subordinate to the repository naming context.

29. (previously presented) The computer system of claim 25 wherein the prefix naming context further includes:

at least one leaf node defined by an interface definition object.

30. (previously presented) The computer system of claim 25, wherein the prefix naming context is defined by a prefix object.

31. (cancelled)

32. (previously presented) A computer system that resolves name collisions by providing type support for multiple type definitions, comprising:

an interface repository including:

a repository naming context; and

a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least one interface definition language declaration, the prefix naming context being adapted to resolve names subordinate to the repository naming context; and

an interface repository loader that accepts as input parameters a specified interface definition language file containing at least one interface definition language declaration, and a specified prefix name, and installs the at least one interface definition language declaration in a prefix naming context having the prefix naming context in the interface repository, and wherein the interface repository loader creates a data file

identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context.

33. (previously presented) A computer system that resolves name collisions by providing type support for multiple type definitions, comprising:

an interface repository including:

a repository naming context; and

a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least one interface definition language declaration, the prefix naming context being adapted to resolve names subordinate to the repository naming context; and

an interface repository loader that accepts as input parameters a specified interface definition language file containing at least one interface definition language declaration, and a specified prefix name, and installs the at least one interface definition language declaration in a prefix naming context having the prefix naming context in the interface repository, and wherein the interface repository loader creates the specified prefix naming context in the interface repository if the specified prefix naming context does not exist therein.

34. (previously presented) The computer system of claim 33, further comprising:

a memory device that stores the interface repository; and

a processing unit that executes operations of the interface repository loader.

35. (previously presented) A computer system that resolves name collisions by providing type support for multiple type definitions, comprising:

an interface repository including:

a repository naming context; and

a prefix naming context subordinate to the repository naming context, the prefix naming context serving as a root naming context for at least

one interface definition language declaration, the prefix naming context being adapted to resolve names subordinate to the repository naming context;

an interface repository loader that accepts as input parameters a specified interface definition language file containing at least one interface definition language declaration, and a specified prefix name, and installs the at least one interface definition language declaration in a prefix naming context having the prefix naming context in the interface repository;

a memory device that stores the interface repository; and

a processing unit that executes operations of the interface repository loader, and further executes the interface repository loader to create a data file identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context.

36. (currently amended) A method of resolving name collisions by providing type support for multiple type definitions, comprising the steps of:

defining in an interface repository a prefix naming context, the prefix naming context being adapted to resolve names subordinate to the repository naming context; and

storing the prefix naming context subordinate to the repository naming context in the interface repository, the prefix naming context forming an interface definition language root context for interface definition objects subordinate to the prefix naming context;

specifying an interface definition language file containing at least one interface definition language declaration;

specifying the prefix naming context; and

storing each interface definition language declaration in the specified interface definition language file into the specified prefix naming context.

37. (previously presented) The method of claim 36, wherein each prefix naming context is stored immediately subordinate to the repository naming context.

38. (cancelled)

39. (currently amended) The method of claim ~~38~~ 36, wherein the step of storing each interface definition language declaration further comprises the steps of:

- creating an interface definition object for the interface definition language declaration;
- storing the interface definition object in the specified prefix naming context; and
- providing the interface definition object with a fully scoped object name including a prefix name from the prefix naming context in which the interface definition object is stored.

B<sup>2</sup>  
40. (previously presented) A method of resolving name collisions by providing type support for multiple type definitions, comprising the steps of:

- defining in an interface repository a prefix naming context, the prefix naming context being adapted to resolve names subordinate to a repository naming context;
- storing the prefix naming context subordinate to the repository naming context in the interface repository, the prefix naming context forming an interface definition language root context for interface definition objects subordinate to the prefix naming context;
- specifying an interface definition language file containing at least one interface definition language declaration;
- specifying a prefix naming context;
- storing each interface definition language declaration in the specified interface definition language file into the specified prefix naming context; and
- creating a data file identified as related to the specified interface definition language file, and containing an identification of the specified prefix naming context.



41. (cancelled)

42. (new) A computer system that resolves name, comprising:

- an interface definition language file containing a first set of one or more interface definitions, wherein the one or more interface definitions include an IDL root context;
- an interface repository naming context;
- an interface repository loader operable to install the one or more interface definitions in the interface repository naming context under a specified prefix naming context, wherein the prefix naming context acts as the root IDL naming context.

43. (new) The computer system of claim 42, further comprising:

- a second interface definition language file containing a second set of one or more interface definitions, wherein the interface definition includes the IDL root context,
- wherein the interface repository loader is further operable to install the second set of one or more interface definitions in the interface repository naming context under a second specified prefix naming context, wherein the second prefix naming context acts as the root IDL naming context.

44. (new) The system of claim 42 wherein the prefix naming context further includes at least one naming context defined by an interface definition object and subordinate to the prefix naming context.

45. (new) The computer system of claim 44 wherein at least one interface definition object has a fully scoped object name including a prefix name of the prefix naming context to which the interface definition object is subordinated.

46. (new) The computer system of claim 42 wherein the prefix naming context is immediately subordinate to the repository naming context.

47. (new) The computer system of claim 42 wherein the prefix naming context further includes at least one leaf node defined by an interface definition object.

48. (new) The computer system of claim 42, wherein the prefix naming context is defined by a prefix object.

49. (new) The computer system of claim 42, wherein the interface repository loader creates a data file identified as related to the interface definition language file, and containing an identification of the specified prefix naming context.

50. (new) The computer system of claim 42, wherein the interface repository loader creates the specified prefix naming context in the interface repository if the specified prefix naming context does not exist therein.

---